REMARKS

Claims 1, 8, 14 and 16 have been amended. Claims 19-23 have been added. Support for the amendments and new claims is found at page 7, lines 2-4 of the present specification. In claim 14, the word "validating" was changed to --invalidating-- to cure a typographical error. Support for this amendment is found at page 21, lines 30 to page 22, line 3 and page 22, lines 23 to page 23, line 2 of the present specification. No new matter has been added. Accordingly, Claims 1-23 are pending in this application and are submitted for consideration.

Applicants' representative thanks the Examiner for taking the time to conduct a personal interview on April 2, 2002.

Applicants acknowledge that claims 17-18 were found to contain allowable subject matter. Since all rejections are addressed herein, Applicants request that claims 1-23 be allowed.

In the Office Action dated November 8, 2001, claims 1-16 were rejected under 35 USC § 103(a) as being unpatentable over AAPA (JP 3-191427) in view of Hoskins (U.S. Patent No. 5,872,978). Applicants submit that claims 1-16 and 19-23 recite subject matter which is neither shown nor suggested by any combination of the cited prior art.

The present invention as defined by the claims is directed to systems and methods for prefetching instructions and data of a program stored in a memory in accordance with at least one instruction address or data address that is part of a pseudo instruction arranged before a predetermined instructions. The instruction address indicates the jump destination of the predetermined instruction and the data

address indicates the address of the data that is necessary for the predetermined instruction. By using the pseudo instruction including the instruction address or the data address, the predetermined instruction is acquired from the buffer without encountering mis-hit.

Independent claims 1, 8 14 and 16 have been amended to clarify the difference between APPA and Hoskins, and the claimed invention. In particular, the pseudo instruction of the claimed invention is not executed by the instruction execution unit. In contrast, the prefetched instruction of APPA is a branch instruction that is executed by the instruction execution unit and is not the same as the pseudo instruction of the claimed invention. Furthermore, the pseudo instruction of Hoskins does not include instruction address or data address for a predetermined instruction follows the pseudo instruction. Furthermore, Hoskins does not disclose that the instruction address or data address is stored before the predetermined instruction is fetched.

Hoskins is directed to an apparatus for translation of program data into machine code format. The apparatus includes an encoder compiler (18) that generates and inserts in a common format code stream a number of pseudo-instructions. Hoskins appears to teach that the pseudo-instructions provide guidance to a decoder translator (24) aimed at improving the speed of economy of translation to the native code. In other words, the pseudo-instructions are used to efficiently perform translation of the program data. For example, the pseudo-instruction (BRANCH_LIKELY) notifies the translator (24) that the conditional branch instruction will follow the pseudo-instruction. The translator (24) translates the branch instruction *in accordance* with the pseudo-instruction.

In contrast, the pseudo instruction of the present invention not only indicates that the branch instruction follows the pseudo instruction, but includes the instruction address or the data address. Accordingly, the branch instruction can be prefetched from the memory before the branch instruction is executed. Furthermore, the claimed pseudo instruction is not executed and is different from the cited prior art. When execution of the instruction is determined, the branch instruction is already stored in the buffer, therefore, it is possible to continuously execute branch instructions without encountering mis-hit.

Since Hoskins does not teach or suggest prefetching an instruction using the pseudo instruction that includes the instruction address or the data address, Hoskins cannot continuously execute the branch instructions to improve the processing efficiently. Furthermore, the pseudo instruction of Hoskins does not include instruction address or data address for a predetermined instruction follows the pseudo instruction. Furthermore, Hoskins does not disclose that the instruction address or data address is stored before the predetermined instruction is fetched.

AAPA (JP 3-191427) also does not teach or suggest using the pseudo instruction that includes the instruction address or the data address. The instruction decoder (10) of AAPA determines whether first data stored in the FIFO memory (9) is a branch instruction and provides second data (branch destination absolute address or relative address) stored in the FIFO memory immediately after the branch instruction with the branch destination address generating circuit (11) when the first data is the branch instruction. The generating circuit (11) provides the branch destination address with the cash controller (2) providing part of branch destination address information with the

cash tag memory (5) to determine cash hit or mis-hit. As described above, AAPA prepares a branch destination address in the cash memory (5) after a branch instruction is read from the cash memory and before the branch instruction is executed. Accordingly, AAPA does not provide a pseudo instruction including a instruction address or a data address before reading a branch instruction, as defined by the claimed invention. In other words, AAPA performs a branch prediction after reading a branch instruction.

Thus, in view of the above, no combination of the APPA and Hoskins shows or suggests each and every element of claims 1, 8 14 and 16, and all intervening claims by their dependencies thereon. Accordingly, Applicants request that the rejection be withdrawn and claims 1-23 be allowed.

In view of the above remarks, the Applicants respectfully submit that each of Claims 1-23 recite subject matter which is neither disclosed nor suggested in the cited prior art. Applicants submit that this subject matter is more than sufficient to render the claimed invention unobvious to a person of ordinary skill in the art. Applicants therefore request that each of 1-23 be found allowable, and this application passed to issue.

If for any reason the Examiner determines that the application is not now in condition for allowance, it is respectfully requested that the Examiner contact, by telephone, the Applicants' undersigned attorney at the indicated telephone number to arrange for an interview to expedite the disposition of this application.

In the event this paper is not timely filed, the Applicants respectfully petitions for an appropriate extension of time. Any fees for such an extension together with any additional fees may be charged to Counsel's Deposit Account No. 01-2300.

Respectfully submitted,

Brian A. Tollefson
Registration No. 46,338

ARENT FOX KINTNER PLOTKIN & KAHN, PLLC
1050 Connecticut Avenue, N.W.,
Suite 400
Washington, D.C. 20036-5339
Tel: (202) 857-6000
Fax: (202) 638-4810

BAT:tdd

MARKED UP COPY OF AMENDED CLAIMS

1. (Amended) A method for prefetching instructions and data of a program stored in a memory, wherein the program includes a pseudo instruction and at least one of an unconditional branch instruction, a conditional branch instruction, a CALL instruction, and a data calling instruction, the pseudo instruction being arranged before the at least one instruction and indicating that the at least one instruction or data follows the pseudo instruction, at least one instruction address [of] or data address being part of the pseudo instruction, wherein the pseudo instruction is not executed, the method comprising the steps of:

reading the program from the memory;

detecting the pseudo instruction;

prefetching the instruction or data from the memory in accordance with the at least one instruction address or the data address; and

storing the prefetched instruction or data in a buffer.

8. (Amended) A microcontroller, comprising:

a buffer, connected to a memory, for storing instructions and data of a program prefetched from the memory, wherein the program includes a pseudo instruction, at least one of an unconditional branch instruction, a conditional branch instruction, a CALL instruction, and a data calling instruction, the pseudo instruction being arranged before the at least one instruction and indicating that the at least one instruction or data follows the pseudo instruction, and at least one instruction address or

data address being part of he pseudo instruction, and wherein the pseudo instruction is not executed;

an instruction execution unit, connected to the buffer, for receiving the instruction and data from the buffer and executing a predetermined processing operation using the instruction and data;

a pseudo instruction detection unit, connected to the memory, for detecting the pseudo instruction included in the program prefetched from the memory; and

an address control unit, connected to the external memory and the pseudo instruction detection unit, for prefetching the instruction or data in accordance with at least one instruction address or data address when the pseudo instruction is detected.

14. (Amended) A device for detecting a pseudo instruction present before a specific instruction, wherein the pseudo instruction includes an opcode and an operand, and wherein the pseudo instruction is not executed, the device comprising:

a detection circuit, connected to a data line, for receiving the pseudo instruction transferred on the data line and detecting the opcode included in the pseudo instruction; and

a detection timing circuit, connected to the detection circuit, for calculating instruction length or the number of operands of the pseudo instruction from the opcode and determining the transfer period of the opcode based on the instruction length or the number of operands, wherein the detection timing circuit supplies a signal for [validating] invalidating the opcode detection operation during an operand transfer period.

15. (Amended) A recording medium having a program stored thereon, wherein the program includes a pseudo instruction and at least one of an unconditional branch instruction, a conditional branch instruction, a CALL instruction, and a data calling instruction, the pseudo instruction being arranged before the at least one instruction and indicating that the at least one instruction or data follows the pseudo instruction, wherein the pseudo instruction is not executed, and wherein the at least one instruction address or data address is part of the pseudo instruction.

16. (Amended) A microcontroller connected to a memory which stores instructions and data, the microcontroller comprising:

an instruction execution unit for reading instructions and data from the memory and processing the read instructions; and

a prefetch circuit unit that receives instructions and data read from the memory in response to a fetch signal, and detects pseudo instructions included in the instructions and data, wherein a pseudo instruction precedes a branch instruction and indicates the existence of the branch instruction, and wherein the pseudo instruction is not executed;

wherein the prefetch circuit unit includes,

a prefetch buffer connected between the instruction execution unit and the memory for temporarily storing instructions and data being transferred from the memory to the instruction execution unit,

a bus interconnecting the prefetch buffer and the memory,

a pseudo instruction detection unit connected to the bus for detecting pseudo instructions among the instructions and data being transferred from the memory to the prefetch buffer,

a holding circuit, connected to the bus and to the pseudo instruction detection unit, for storing operands of the pseudo instruction,

a pseudo instruction buffer for temporarily storing instructions and data fetched from a location in the memory which is pointed to by the branch instruction following the pseudo instruction,

an address control unit for generating the fetch signal and for generating a memory address which points to the address of a next word to be read from the memory, and wherein when the pseudo instruction detection unit detects a pseudo instruction, the instructions and data pointed to by the pseudo instruction are fetched from the memory by the address control unit and stored in the pseudo instruction buffer so that when the branch instruction following the pseudo instruction is processed by the instruction execution unit, if the branch is taken, the instructions and data pointed to by the branch instruction have been prefetched and stored in the pseudo instruction buffer.